PORTAL USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ◉ The ACM Digital Library ○ The Guide

efficient locking for concurrent operations on b-trees

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
efficient locking for concurrent operations on b trees

Found **55,896** of **171,143**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Efficient locking for concurrent operations on B-trees

Philip L. Lehman, s. Bing Yao

December 1981 **ACM Transactions on Database Systems (TODS)**, Volume 6 Issue 4

Publisher: ACM Press

Full text available: pdf(1.40 MB) Additional Information: full citation, abstract, references, citings, index terms

The B-tree and its variants have been found to be highly useful (both theoretically and in practice) for storing large amounts of information, especially on secondary storage devices. We examine the problem of overcoming the inherent difficulty of concurrent operations on such structures, using a practical storage model. A single additional "link" pointer in each node allows a process to easily recover from tree modifications performed by other concurrent processes. Our solution ...

Keywords: B-tree, concurrenct algorithms, concurrency controls, consistencey, correctness, data structures, database, index organizations, locking protocols, multiway search trees

2 Operation specific locking in B-trees

A. Biliris

June 1987 **Proceedings of the sixth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Publisher: ACM Press

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings, index terms

B-trees have been used as an access and for both primary and secondary indexing for quite some time. This paper presents a deadlock free locking mechanism in which different processes make use of different lock types in order to reach the leaf nodes. The compatibility relations among locks on a node, do not exclusively depend on their type, but also on the node status and the number and kind of processes acting currently on the node. As a result, a number of insertion or deletion processes ...

3 Algorithms and data structures: Concurrent cache-oblivious b-trees

Michael A. Bender, Jeremy T. Fineman, Seth Gilbert, Bradley C. Kuszmaul

July 2005 **Proceedings of the 17th annual ACM symposium on Parallelism in algorithms and architectures SPAA'05**

Publisher: ACM Press

Full text available: pdf(180.51 KB) Additional Information: full citation, abstract, references, index terms

This paper presents concurrent cache-oblivious (CO) B-trees. We extend the cache-oblivious model to a parallel or distributed setting and present three concurrent CO B-trees. Our first data structure is a concurrent lock-based exponential CO B-tree. This data structure supports insertions and non-blocking searches/successor queries. The second and third data structures are lock-based and lock-free variations, respectively, on the packed-memory CO B-tree. These data structures support range queri ...

Keywords: cache-oblivious b-tree, concurrent b-tree, exponential tree, lock free, non-blocking, packed-memory array

4 A symmetric concurrent B-tree algorithm

Vladimir Lanin, Dennis Shasha

November 1986 **Proceedings of 1986 ACM Fall joint computer conference**

Publisher: IEEE Computer Society Press

Full text available: pdf(1.10 MB) Additional Information: full citation, references, citings, index terms

5 A framework for the performance analysis of concurrent B-tree algorithms

Theodore Johnson, Dennis Shasha

April 1990 **Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Publisher: ACM Press

Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, citings, index terms

Many concurrent B-tree algorithms have been proposed, but they have not yet been satisfactorily analyzed. When transaction processing systems require high levels of concurrency, a restrictive serialization technique on the B-tree index can cause a bottleneck. In this paper, we present a framework for constructing analytical performance models of concurrent B-tree algorithms. The models can predict the response time and maximum throughput. We analyze three algorithms: Naive Lock-coupling, Op ...

6 Locking without blocking: making lock based concurrent data structure algorithms nonblocking

John Turek, Dennis Shasha, Sundeep Prakash

July 1992 **Proceedings of the eleventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Publisher: ACM Press

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

Nonblocking algorithms for concurrent data structures guarantee that a data structure is always accessible. This is in contrast to blocking algorithms in which a slow or halted process can render part or all of the data structure inaccessible to other processes. This paper proposes a technique that can convert most existing lock-based blocking data structure algorithms into nonblocking algorithms with the same functionality. Our instruction-by-instruction transformation can be ap ...

7 Concurrent search structure algorithms

Dennis Shasha, Nathan Goodman

March 1988 **ACM Transactions on Database Systems (TODS)**, Volume 13 Issue 1

Publisher: ACM Press

Full text available: pdf(2.72 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A dictionary is an abstract data type supporting the actions member, insert, and delete. A search structure is a data structure used to implement a dictionary. Examples include B trees, hash structures, and unordered lists. Concurrent algorithms on search structures

can achieve more parallelism than standard concurrency control methods would suggest, by exploiting the fact that many different search structure states represent one dictionary state. We present a framework for verifying such a ...

8 The performance of current B-tree algorithms

Theodore Johnson, Dennis Sasha

March 1993 **ACM Transactions on Database Systems (TODS)**, Volume 18 Issue 1

Publisher: ACM Press

Full text available: pdf(2.87 MB) Additional Information: full citation, references, citings, index terms, review

Keywords: B-trees, concurrent B-trees, concurrent data structures, performance of concurrent algorithms

9 Concurrent set manipulation without locking

Vladimir Lanin, Dennis Shasha

March 1988 **Proceedings of the seventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Publisher: ACM Press

Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings

Set manipulation consists of the actions insert, delete, and member on keys. We propose a concurrent set manipulation algorithm that uses no locking at all and requires no aborts, relying instead on atomic read-modify-write operations on single (data) locations. The algorithm satisfies order-preserving serializability through conditions that are strictly looser than existing algorithms

10 Concurrent manipulation of binary search trees

H. T. Kung, Philip L. Lehman

September 1980 **ACM Transactions on Database Systems (TODS)**, Volume 5 Issue 3

Publisher: ACM Press

Full text available: pdf(1.95 MB) Additional Information: full citation, abstract, references, citings, index terms

The concurrent manipulation of a binary search tree is considered in this paper. The systems presented can support any number of concurrent processes which perform searching, insertion, deletion, and rotation (reorganization) on the tree, but allow any process to lock only a constant number of nodes at any time. Also, in the systems, searches are essentially never blocked. The concurrency control techniques introduced in the paper include the use of special nodes and pointers to redirect se ...

Keywords: binary search trees, concurrency controls, concurrent algorithm, consistency, correctness, data structures, databases, locking protocols

11 Performance of B-tree concurrency control algorithms

V. Srinivasan, Michael J. Carey

April 1991 **ACM SIGMOD Record , Proceedings of the 1991 ACM SIGMOD international conference on Management of data SIGMOD '91**, Volume 20 Issue 2

Publisher: ACM Press

Full text available: pdf(1.09 MB) Additional Information: full citation, references, citings, index terms

12 Efficient concurrency control in multidimensional access methods

Kaushik Chakrabarti, Sharad Mehrotra

June 1999 **ACM SIGMOD Record , Proceedings of the 1999 ACM SIGMOD international**

conference on Management of data SIGMOD '99, Volume 28 Issue 2
Publisher: ACM Press

Full text available: pdf(1.79 MB) Additional Information: full citation, abstract, references, citings, index terms

The importance of multidimensional index structures to numerous emerging database applications is well established. However, before these index structures can be supported as access methods (AMs) in a "commercial-strength" database management system (DBMS), efficient techniques to provide transactional access to data via the index structure must be developed. Concurrent accesses to data via index structures introduce the problem of protecting ranges specified in the retrieval fr ...

13 Highly concurrent cache consistency for indices in client-server database systems
Markos Zaharioudakis, Michael J. Carey
June 1997 **ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data SIGMOD '97**, Volume 26 Issue 2
Publisher: ACM Press

Full text available: pdf(1.81 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we present four approaches to providing highly concurrent B+-tree indices in the context of a data-shipping, client-server OODBMS architecture. The first performs all index operations at the server, while the other approaches support varying degrees of client caching and usage of index pages. We have implemented the four approaches, as well as the 2PL approach, in the context of the SHORE OODB system at Wisconsin, and we present experimen ...

14 Adaptable concurrency control for atomic data types
M. S. Atkins, M. Y. Coady
August 1992 **ACM Transactions on Computer Systems (TOCS)**, Volume 10 Issue 3
Publisher: ACM Press

Full text available: pdf(2.54 MB) Additional Information: full citation, abstract, references, index terms, review

In many distributed systems concurrent access is required to a shared object, where abstract object servers may incorporate type-specific properties to define consistency requirements. Each operation and its outcome is treated as an event, and conflicts may occur between different event types. Hence concurrency control and synchronization are required at the granularity of conflicting event types. With such a fine granularity of locking, the occurrence of conflicts is likely to be lower tha ...

Keywords: concurrent access to shared data, hybrid locking, optimistic locking, pessimistic locking, transactions serializability

15 Session 3: Extendible hashing for concurrent operations and distributed data
Carla Schlatter Ellis
March 1983 **Proceedings of the 2nd ACM SIGACT-SIGMOD symposium on Principles of database systems**
Publisher: ACM Press

Full text available: pdf(1.14 MB) Additional Information: full citation, abstract, references, citings

The extendible hash file is a dynamic data structure that is an alternative to B-trees for use as a database index. While there have been many algorithms proposed to allow concurrent access to B trees similar solutions for extendible hash files have not appeared. In this paper, we present solutions to allow for concurrency that are based on locking protocols and minor modifications in the data structure.Another question that deserves consideration is whether these indexing structures can be adap ...

16 A methodology for implementing highly concurrent data objects

Maurice Herlihy
November 1993 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 15 Issue 5
Publisher: ACM Press

Full text available: pdf(1.60 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A concurrent object is a data structure shared by concurrent processes. Conventional techniques for implementing concurrent objects typically rely on critical sections; ensuring that only one process at a time can operate on the object. Nevertheless, critical sections are poorly suited for asynchronous systems: if one process is halted or delayed in a critical section, other, nonfaulty processes will be unable to progress. By contrast, a concurrent object i ...

17 Performance of B± tree concurrency control algorithms
V. Srinivasan, Michael J. Carey
October 1993 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 2 Issue 4
Publisher: Springer-Verlag New York, Inc.

Full text available: pdf(2.67 MB) Additional Information: full citation, abstract, references, citings

A number of algorithms have been proposed to access B+-trees concurrently, but they are not well understood. In this article, we study the performance of various B+-tree concurrency control algorithms using a detailed simulation model of B+-tree operations in a centralized DBMS. Our study covers a wide range of data contention situations and resource conditions. In addition, based on the performance of the set of B+-tree concurrency control algorithms, ...

Keywords: B+-tree structures, data contention, lock modes, performance, resource conditions, simulation models, workload parameters

18 Concurrent operations on extendible hashing and its performance
Vijay Kumar
June 1990 **Communications of the ACM**, Volume 33 Issue 6
Publisher: ACM Press

Full text available: pdf(1.35 MB) Additional Information: full citation, abstract, references, index terms, review

Extendible hashing is a dynamic data structure which accommodates expansion and contraction of any stored data efficiently. In this article, an algorithm has been developed for managing concurrent operations on extendible hashing by achieving optimal memory utilization by supporting directly expansion and contraction, page split, and merge. The results of this study have been encouraging in the sense that it seems to provide a higher degree of concurrency compared to other algorithms on an ...

Keywords: atomic, blocking, global depth, local depth, pseudo key, roll back, verification process

19 Session 2: Concurrency control mechanisms and the serializability of concurrent tree algorithms
Ray Ford, Jim Calhoun
April 1984 **Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems**
Publisher: ACM Press

Full text available: pdf(814.98 KB) Additional Information: full citation, references, citings

20 Linearizability: a correctness condition for concurrent objects

Maurice P. Herlihy, Jeannette M. Wing

July 1990 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 12 Issue 3
Publisher: ACM Press

Full text available: pdf(2.21 MB) Additional Information: full citation, abstract, references, citings, index terms, review

A concurrent object is a data object shared by concurrent processes. Linearizability is a correctness condition for concurrent objects that exploits the semantics of abstract data types. It permits a high degree of concurrency, yet it permits programmers to specify and reason about concurrent objects using known techniques from the sequential domain. Linearizability provides the illusion that each operation applied by concurrent processes takes effect instantaneously at some point between i ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player